

REMARKS

Claims 1-39 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 103(a) Rejection:

The Office Action rejected claims 1-39 under 35 U.S.C. § 103(a) as being unpatentable over Barker, et al. (U.S. Patent 6,363,421) (hereinafter "Barker") in view of Sampat, et al. (U.S. Patent 6,279,029) (hereinafter "Sampat"). Applicants respectfully traverse this rejection in light of the following remarks.

Regarding claim 1, contrary to the Examiner's assertion, Barker in view of Sampat does not teach or suggest that the event gateway comprises a plurality of **event distribution server sinks configured to receive events generated by the managed objects and distribute the events to the one or more managers** such that one of the managers receives events from a plurality of different ones of the event distribution server sinks.

Barker discloses a method for remotely managing a plurality of network elements of a telecommunications network through a special communications link. A management computer is connected to an element management system server 32 through a communication link including the internet. At least one of the network elements 14 is also coupled to the element management server 32 through the internet and is managed via communications conveyed through the element management server 32 between the management computer and the at least one network element 14. Sampat teaches a client/server system that sends and received multicast media data streams over a network.

The Examiner admits that Barker "does not disclose a plurality of server sinks configured to receive events generated by the managed objects and distribute the events to one or more managers" and asserts that Sampat teaches such a plurality of sinks.

Applicants respectfully disagree with the Examiner's characterization of Sampat and submit that Sampat does not teach or suggest event distribution server sinks configured to receive events generated by managed objects and distribute the events to the one or more managers such that one of the managers receives events from a plurality of different one of the event distribution server sinks. Instead, Sampat teaches a client/server architecture for a network based multicast system that includes a media services manager and media services that sends and received multicast media data streams over a network (Sampat, Abstract, column 2, lines 25-36).

Sampat uses source and sink media service providers (MSPs) that assist in the receipt of and playing of data streams, respectively (Sampat, column 9, lines 24). However, Sampat's sink MSPs are not *event distribution* server sinks that receive and distribute events. In contrast, Sampat teaches that sink MSPs are used to transfer received media data (text, video, and audio, for example) to appropriate hardware drivers for local playing. (Sampat, Fig. 16, items 1614, 1618, and 1622, Fig. 18, items 1814, 1414, and 1822). Specifically, Sampat teaches that media server provider sinks are used to play multicast data either received from the network or locally recorded (Sampat, column 14, lines 18-28). Additionally, Sampat describes how "[v]ideo sink MSP 1814 and text sink MSP 1822 receive a video data stream and a text data stream, respectively, from MSM 1808 and transmits the video and text data to display driver ... for display on [a] monitor" (Sampat, column 14, lines 38 – 41). Similarly, Sampat teaches that an audio sink transmits received audio data to an audio driver for playing on audio hardware (Sampat, column 14, lines 42-44). Thus, Sampat teaches the use of media service provider sinks to transfer received media data to appropriate hardware device drivers. The source and sink media service providers of Sampat have nothing to do with receiving and distributing events of any kind, let alone events generated by managed objects and distributed to one or more managers. The Examiner cites column 13, line 62 to column 14, line 57 of Sampat for support. However, as shown above, this passage does not teach event distribution sinks, but instead teaches media service provider sinks that transfer media data to appropriate hardware device drivers.

Additionally, sink MSPs, as taught by Sampat, each receive data from a single, data stream and forward it to a single device driver. Sampat teaches that media service manager (MSM) starts a sink MSP for each data stream (Sampat, column 14, lines 6-11). Further, as shown above, each sink MSP transmits data to a specific device driver (Sampat, column 14, lines 38-44). Thus, Sampat's sink MSPs do not distribute the data that they receive, but merely transfer it from a specific input to a specific output. Sampat teaches that a single process, the media services manager (MSM) actually determines how to route incoming data. For example, Sampat states, "[u]pon receiving new data from the network, the MSM transmits the data to the appropriate MSP." (Sampat, column 15, lines 51-53).

Applicants also submit that the combination of Barker and Sampat as suggested by the Examiner would not result in an event gateway that comprises a plurality of event distribution sinks configured to receive events generated by the managed objects and distribute the events to the one or more managers. Instead any modification based on Sampat of Barker's system for managing network elements to use the sink media service providers of Sampat would at most result in a system for managing media service providers. However, since neither Barker nor Sampat teach the use event distribution sinks (that receive events from managed objects and distribute the events to one or more managers), no combination of Barker and Sampat would suggest such event distribution sinks.

Furthermore, Barker clearly teaches a single event distributor 140 (Barker -- Fig. 4). Barker teaches that this single event distributor "is responsible for filtering and routing of all events in the system" (Barker -- col. 17, lines 5-6). Thus, Barker actually teaches away from a plurality of event distribution server sinks configured such that one of the managers receives events from a plurality of different ones of the event distribution server sinks.

Additionally, Applicants respectfully disagree with the Examiner's assertion that Barker's applications 44 and 50 are managed objects that generate events. Barker clearly

describes applications 44 and 50 as client applications that are distinct from the managed objects. *See, e.g.*, Barker -- col. 11, lines 47-60; col. 13, line 46 – col. 16, line 12; col. 39, line 55 – col. 40, line 16. At the top of Fig. 6, Barker defines an example of a managed object as a network element. Thus, in Fig. 2, the managed objects are represented by network element 14, **not client applications 44 and 50**. Anyone of ordinary skill in the art reading Barker would understand that client applications 44 and 50 are not the managed objects in Barker's system. Furthermore, **there is no teaching in Barker that client applications 44 and 50 generate events**. A careful reading of each section of Barker cited by the Examiner reveals absolutely no mention at all that client applications 44 and 50 generate events received by element management system server 32. Applicants request that the Examiner quote the exact column and line numbers where he believes Barker teaches that client applications 44 and 50 are managed network elements and generate the events described in Barker. **Although Applicants have previously asserted this argument, Applicants note that the Examiner has never supplied any rebuttal of this specific argument.**

Thus, in light of the above remarks, Applicants assert that the rejection of claim 1 is not supported by the cited art and withdrawal of the rejection is respectfully requested. Similar remarks as discussed above in regard to claim 1 apply to claims 9, 14 and 27.

In regard to claim 10, Barker does not teach an event port registry server comprising a plurality of ports and an event port registry, wherein the event port registry server is coupled to the event distribution server, wherein the event ports comprise communication channels for the delivery of the events to the one or more managers, and wherein the event port registry provides information to the event distribution server regarding which ports correspond to which managers. The Examiner cites col. 9, line 22 – col. 10, line 49; and col. 14, line 35 – col. 15, line 32, of Barker in regard to claim 10. However, a careful reading of these sections of Barker reveals absolutely no description at all of anything that corresponds to an event port registry server comprising a plurality of event ports that comprise communication channels for the delivery of the events to the one or more managers. Nor does Barker contain any teaching that corresponds to an

event port registry that provides information to an event distribution server regarding which ports correspond to which managers. **Although Applicants have previously asserted this argument, Applicants note that the Examiner has never supplied any rebuttal of this specific argument.**

Thus, in light of the above remarks, Applicants assert that the rejection of claim 10 is not supported by the cited art and withdrawal of the rejection is respectfully requested. Similar remarks as discussed above in regard to claim 10 apply to claims 23 and 35.

Regarding claim 12, Barker does not teach a plurality of event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers. The Examiner cites col. 29, line 27 – col. 30, line 42; and col. 37, line 4 – col. 38, line 63, of Barker in regard to claim 12. However, these sections of Barker reveal absolutely nothing at all that corresponds to a plurality of event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers. In fact, the first cited section (Barker, col. 29, line 27 – col. 30, line 42) describes how Barker's system provides overload control by limiting the amount of network traffic through only allowing a limited number of client sessions and running applications. This cited section also describes Barker's preferred method of software version management. Neither of these topics discusses load balancing. The Examiner's second cited passage (col. 37, line 4 – col. 38, line 63) describes the major functions of Barker's event handler and goes on to provide overviews of Barker's Network Element Status Table, Network Element Status API, Event Configuration File, Admin Log, and EMAPI, none of which mention (or have anything to do with) a plurality of event distribution server sinks that are distributing to provide load balancing of the events to the one or more managers. Thus, the rejection of claim 12 is not supported by the cited art and withdrawal of the rejection is respectfully requested. **Although Applicants have previously asserted this argument, Applicants note that the Examiner has never supplied any rebuttal of this specific argument.** Similar arguments apply in regard to claims 25 and 37.

CONCLUSION

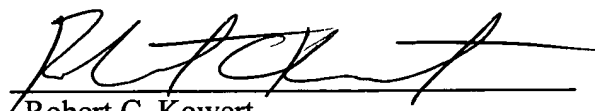
Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-48200/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Fee Authorization Form authorizing a deposit account debit in the amount of \$
for fees ().
- ☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: August 18, 2004